2009年度　修士論文

モバイルマルチキャストにおける
階層的な鍵構造を考慮した鍵管理方式

指導教員　戸川 望 教授

早稲田大学大学院　基幹理工学研究科
情報理工学専攻

5108B029–1

金井 孝仁

2010年2月5日

目 次

第1章

序論

1.1 本論文の背景と意義

近年，インターネットコンテンツの多様化やコンピューティング環境の発展によってインターネット利用者数が激増し，平成 21 年 9 月末現在での FTTH アクセスサービス，DSL アクセスサービスの契約数はそれぞれ約 1651 万人，約 1050 万人を超えており，平成 20 年末現在での世帯，個人でのインターネット普及率はそれぞれ約 91%，約 75%となっている [14]．それに伴い，テレビやラジオ放送など音声や映像を複数の受信者に配信する放送型アプリケーションや，複数の場所で音声や画像などを共有した遠隔会議など，マルチメディアでの需要が増えている．これらのアプリケーションの共通点は「1 つのデータを複数の受信者に配信する」という点である．この 1 対多通信を可能にする通信方式をマルチキャストという．

マルチキャストでは，送信者から送り出されたパケットは必要に応じてルータで複製され，データの受信者へと配信される．ユニキャストでは 1 人の受信者に対し，1 つのパケットが必要になるが，マルチキャストでは複数の受信者に対し，1 つのパケットで良いため，ネットワーク資源を節約できるというメリットがある．しかし，マルチキャストでは UDP（User Datagram Protocol）を用いて通信を行うため，信頼性が確保されないというデメリットがある．そこで，マルチキャストにおいて信頼性を確保した通信，つまり高信頼マルチキャストに関する研究が進められている．この高信頼マルチキャストでは，信頼性，順序保証，セッション管理，セキュリティなどといった様々な要求がある．

本論文では，高信頼マルチキャストの信頼性やセキュリティを保証する技術の 1 つとして，グループ鍵管理に着目した．マルチキャストのグループ鍵管理とは，同じデータを受信するものを 1 つのグループとし，そのグループごとにグループ鍵と呼ばれる鍵を持ち，鍵を用いてデータの暗号化し，またグループメンバーの認証をすることで秘密性を保護する技術である．このグループ鍵はグループメンバーの参加/離脱に伴って，秘密性の保護のために逐一更新される．この鍵配布や鍵更新の処理において，効率的，かつ信頼性を確保した鍵管理を実現する鍵管理手法に関する研究として，鍵更新に要する処理量を減らすことを目的とした方式 [12] や鍵管理サーバの負荷分散を目的とした方式 [5] など様々な研究が進められている．

グループ鍵管理ではメンバーの変更に伴って，グループ鍵が更新されるため，鍵更新に要する処理時間をいかにして低減するかが重要となる．そのため，本論文ではグループ鍵管理におけるグループ鍵の更新に要する処理時間の低減に着目した．鍵更新の処理時間の低減を目的とした既存の鍵管理方式の中でも，モバイルマルチキャストにおいて，特定の場合にグループ鍵の更新を行わない鍵管理方式について分析し，問題点を改善して鍵更新における処理時間を低減できるような鍵管理方式を提案する．提案手法では，1 つのグループを階層化して複数のサブグループを作ることで，鍵更新における処理時間を低減し，無通信時間の低減を図っている．

1.2 本論文の概要

本論文では，モバイルマルチキャストにおけるグループ鍵を考慮した鍵管理方式を提案する．本論文の構成を以下に示す.

第 2 章「Group Key Locking（GKL）**方式による鍵管理とその問題点」**では，既存の鍵管理方式の 1 つとして，グループメンバーの変更が無い場合に，グループ鍵の鍵更新を行わないようにする Group Key Locking（GKL）方式について，モバイルソース及びノードがハンドオフする場合の動作を説明し，処理時間や安全性における問題点を挙げる.

第 3 章「階層的な鍵構造を考慮したグループ鍵管理手法の提案」では，第 2 章で説明した GKL 方式を基にして，グループ鍵の鍵更新における処理時間を低減することを目的とした，階層的な鍵構造を考慮したグループ鍵管理手法を提案する．第 2 章と同様にモバイルソース及びノードがハンドオフする場合の動作について，また提案手法における鍵更新処理等，他の動作についても説明し，提案手法の特徴として考えられる，階層的な鍵構造を用いた分散処理によるメリット，鍵数の増加によるデメリットについて述べる.

第 4 章「提案手法の評価」では，第 3 章で提案した手法に対する評価を行う．具体的には，あるネットワークにおけるハンドオフ処理について，実装した提案手法と GKL 方式においてハンドオフ処理を行った場合の処理時間をシミュレーションによって比較する．その結果をもとに，ハンドオフ処理に要する処理時間について提案手法の有効性を示す．また，提案手法の課題としてグループ鍵サーバの負荷や各ノードでの暗号化・複合化の複雑化について述べる.

第 5 章「結論」では，本論文を総括する.

第2章

Group Key Locking（GKL）方式による鍵管理とその問題点

2.1 本章の概要

本章では，グループ鍵管理方式の中でも，モバイルマルチキャスト環境において，グループメンバーの変更が無い場合に，グループ鍵の鍵更新を防ぐことで鍵更新の処理時間を低減する GKL 方式のアルゴリズムについて説明し，問題点を挙げる．

2.2 GKL 方式

マルチキャストでは，モバイルソースもしくはモバイルノードのハンドオフが起こったときにグループ鍵の更新が発生する．しかし，メンバーシップの変更を伴わないハンドオフの場合においても鍵更新が起こってしまい，結果的にはメンバーが変わっていないにも関わらず，無駄な鍵更新をしてしまうといったことが起こる．GKL 方式では，このような状況での無駄な鍵更新を防ぐことで，鍵更新に要する処理時間を低減することができる．

2.3 チャンネル

GKL 方式を説明する上で,（S,G）という表記について説明する [3]．S はマルチキャストソースアドレス，G はマルチキャストグループアドレスを意味し,（S,G）はチャンネルと呼ばれている．

チャンネルのグループ鍵がチャンネルの全てのメンバーと共有される．GKL 方式では，メンバーシップの変更を伴わないハンドオフの場合において，グループ鍵の鍵更新を防ぐ方式なので，モバイルソースとモバイルノードが，同じマルチキャストグループ鍵管理ドメインの中を移動する場合について考える．

2.4 GKL 方式の動作

GKL 方式では，モバイルソースのハンドオフが起こるとき，ソースアドレスは oCoA（old Care of Address）から nCoA（new Care of Adress）に変えられる．チャンネル移動の間，メンバーシップ変更が無ければ,（oCoA,G）のメンバーと（nCoA,G）のメンバーは同じことになる．よって,（oCoA,G）のグループ鍵は（nCoA,G）に再利用できるということになる．モバイルソースのハンドオフの例を図 2.1 に示す.

また，モバイルノードが新しい FN（Foreign Network）へ移動するときハンドオフは起こり，モバイルノードは FN の MR（Multicast Router）を通してチャンネル（S,G）に加わり，前のチャンネル (S,G) を去る．その結果,（S,G）のメンバーは変わらないので,（S,G）のグループ鍵は再利用できるということになる．モバイルノードのハンドオフの例を図 2.2 に示す.

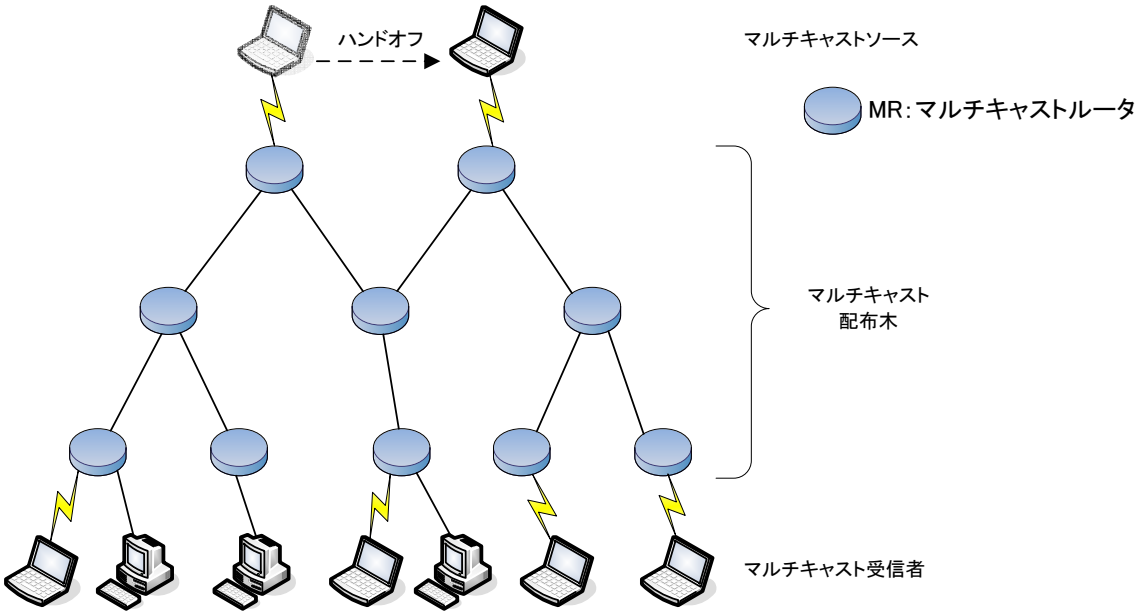

図 2.1: モバイルソースのハンドオフ.

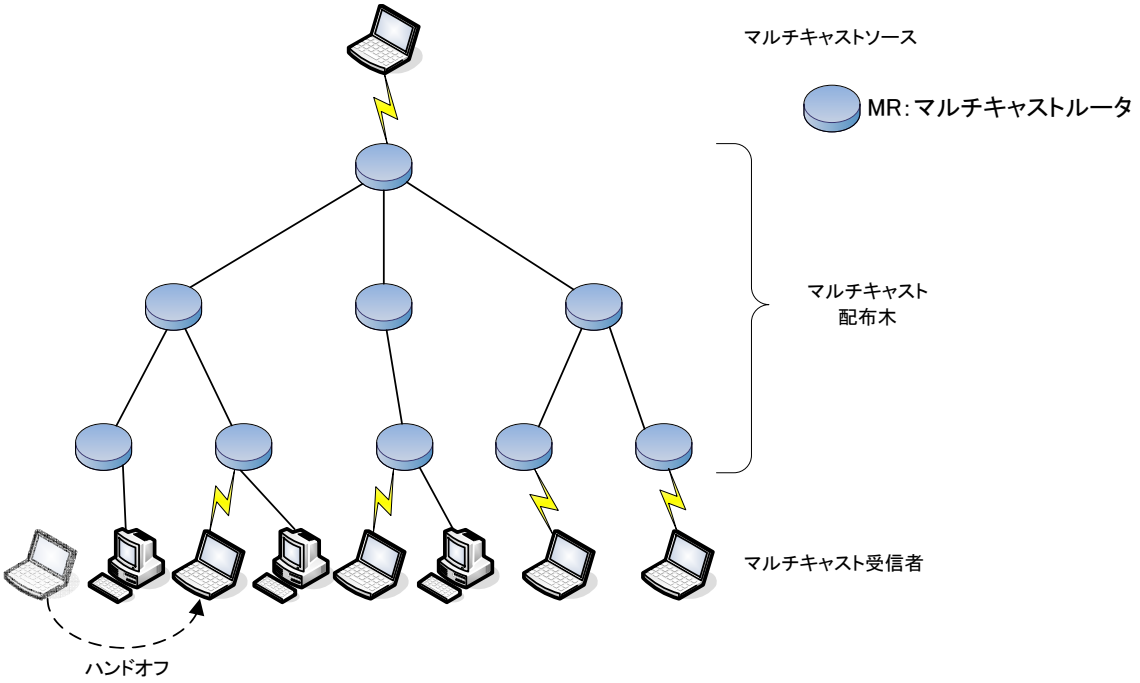

図 2.2: モバイルノードのハンドオフ.

2.5 制御メッセージ

GKL 方式の具体的な動作を説明する前に，まず GKL 方式で用いられる制御メッセージについて説明する．

モバイルソースマルチキャストとして，Mobile SSM[7] を用いたモバイルマルチキャスト環境を仮定する．GKL 方式は，MSSM で定義された MLD（Multicast Listener Discovery）プロトコルを用いた制御メッセージを使用する．このメッセージはモバイルソースからノードに送られ，グループメンバーシップに関連付けられる．また，GKL 方式で利用する GKL Source Message，GKL Done Message，GKL Ack Message，GKL Receiver Message という新しい制御メッセージを用いる．

GKL Source Message

このメッセージはハンドオフするモバイルソースが nCoA を設定するとき，Fast Handover[8] のようなプロトコルを用いてグループ鍵サーバに送られる．このメッセージが持つ情報には，モバイルソースの oCoA と nCoA，グループアドレス G,（oCoA,G）のグループ鍵 K が含まれる．グループ鍵サーバがこのメッセージに応答すると，GKL Done Message の応答までの間，チャンネル（oCoA,G）と（nCoA,G）のグループ鍵更新をロックする．

GKL Done Message

このメッセージはチャンネル移動が終わったとき，モバイルソースによってグループ鍵サーバに送られる．このメッセージには，GKL Source Message の情報から（oCoA,G）を除外したものが含まれる．グループ鍵サーバがこのメッセージに応答すると，チャンネル（oCoA,G）と（nCoA,G）のグループ鍵更新のロックを解く．

GKL Ack Message

このメッセージは，グループ鍵サーバが GKL Source Message や GKL Done Message，GKL Receiver Message のようなメッセージを受け取ったことを通知するために，グループ鍵サーバによってモバイルソース及びノードに送られる．

GKL Receiver Message

このメッセージはハンドオフするモバイルノードが nMR（new Multicast Router）を設定するとき，Fast Handover のようなプロトコルを用いて，（S,G）のメンバーであるモバイルノードによってグループ鍵サーバに送られる．このメッセージが持つ情報には，S，G，モバ

イルノードの oMR（old Multicast Router）と nMR，K が含まれる．グループ鍵サーバがこのメッセージに応答すると，K の鍵更新をロックする．

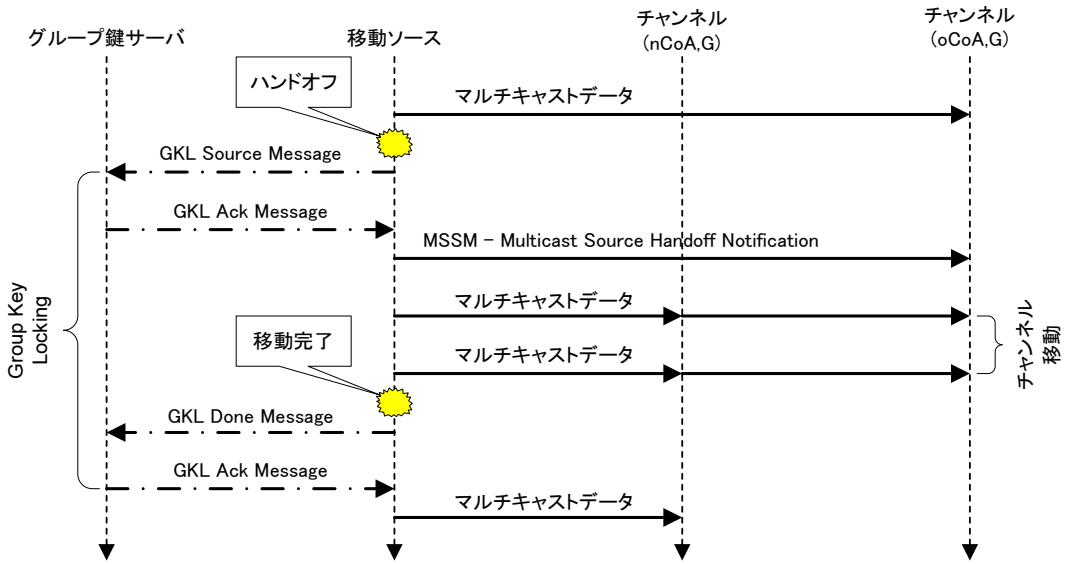

図 2.3: モバイルソースのハンドオフにおける GKL のプロセス.

2.6 モバイルソースのハンドオフ

モバイルソースがハンドオフした場合について，鍵更新が起こる場合の処理と GKL 方式を用いた場合の処理を説明する.

2.6.1 鍵更新が発生する場合

処理フローは以下の通りである.

1：モバイルソースのハンドオフが起こるとき，モバイルソースは nCoA を設定する.

2：モバイルソースがチャンネル（oCoA,G）から（nCoA,G）に移動する.

3：モバイルソースのハンドオフに伴い，チャンネル（nCoA,G）の新しいグループ鍵を設定し，グループメンバーに送信する.

この場合，モバイルソースはハンドオフしたが，結果的にハンドオフ前後でチャンネル（oCoA,G）と（nCoA,G）のグループメンバーは同じであるにも関わらず，グループ鍵更新が起こっている．よって，ここで起こったグループ鍵更新は無駄な処理であるといえる.

2.6.2 GKL 方式を用いた場合

GKL 方式は，2.5 節で説明した制御メッセージを用いたグループ鍵管理である．モバイルソースにおける GKL のプロセスを図 2.3 に示す．処理フローは以下の通りである.

1：モバイルソースのハンドオフが起こるとき，モバイルソースは nCoA を設定する.

2： モバイルソースは，GKL Source Message をグループ鍵サーバに送信する．

3： グループ鍵サーバが GKL Source Message に応答すると，（oCoA,G）と（nCoA,G）のグループ鍵更新をロックし，モバイルソースに GKL Ack Message を送る．

4： GKL Ack Message を受けた後，モバイルソースは Multicast Source Handoff Notification をチャンネル（oCoA,G）に送る．

5： Multicast Source Handoff Notification に基づいて，チャンネル（oCoA,G）から（nCoA,G）に移動する．モバイルソースは，（oCoA,G）と（nCoA, G）の両方にマルチキャストデータを送ることができる．

6： 移動が完了すると，モバイルソースは GKL Done Message をグループ鍵サーバに送信する．

7： グループ鍵サーバは，（oCoA,G）と（nCoA, G）のグループ鍵更新のロックを解き，モバイルソースに GKL Ack Message を送る．このとき，（oCoA,G）と（nCoA, G）は異なるチャンネルだが同じノードであり，同じ鍵である．

モバイルソースのハンドオフにおいて，ハンドオフによってモバイルソースのアクセスポイントが変わるとき，GKL 方式では上記のプロセスによって通常は起こってしまうグループ鍵更新を防ぐことができ，モバイルソースのハンドオフに伴う処理時間を減らすことができる．

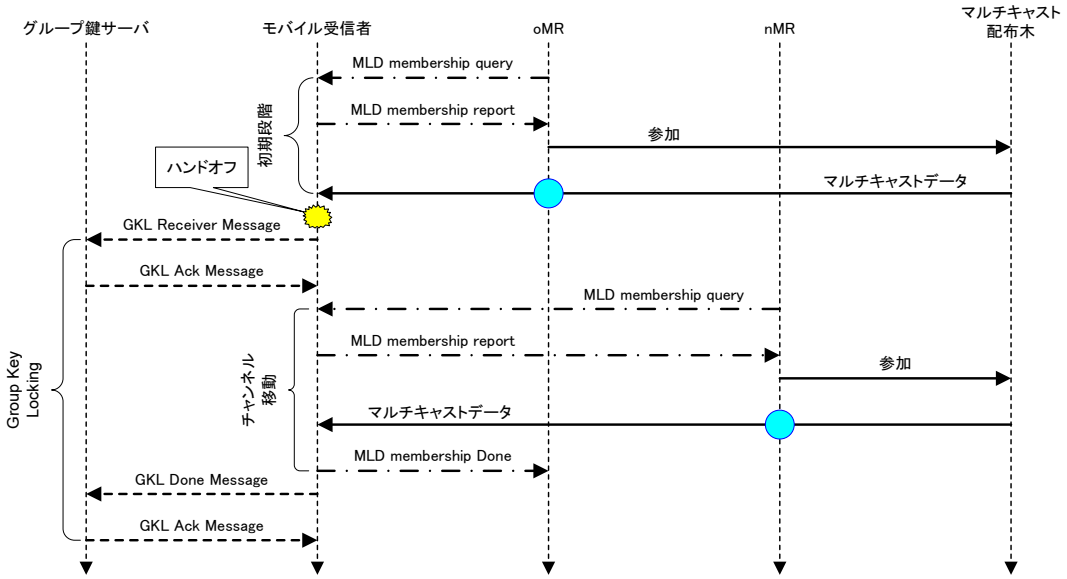

図 2.4: モバイルノードのハンドオフにおける GKL のプロセス.

2.7 モバイルノードのハンドオフ

モバイルノードがハンドオフした場合について，鍵更新が起こる場合の処理と GKL 方式を用いた場合の処理を説明する.

2.7.1 鍵更新が発生する場合

処理フローは以下の通りである.

1：モバイルノードのハンドオフが起こるとき，ノードは nMR を設定する.

2：モバイルノードが一時的にグループから離脱し，グループ鍵の更新が行われ，新たなグループ鍵をグループメンバーに送信する.

3：離脱していたモバイルノードが再びグループに参加し，グループ鍵の更新が行われ，新たなグループ鍵をグループメンバーに送信する.

モバイルノードのハンドオフの場合は，モバイルノードの離脱/参加に伴い，2 回のグループ鍵更新が行われている．しかし，モバイルソースのハンドオフの場合と同様に，結果的にハンドオフ前後でグループメンバーは同じであり，不要なグループ鍵更新が 2 回行われてしまっていることになる.

2.7.2 GKL 方式を用いた場合

モバイルノードにおける GKL のプロセスを図 2.4 に示す．処理フローは以下の通りである.

1：モバイルノードのハンドオフが起こるとき，ノードは nMR を設定する．

2：モバイルノードは，GKL Receiver Message をグループ鍵サーバに送信する．

3：グループ鍵サーバが GKL Receiver Message を受け取ったとき，(S,G）についてのグループ鍵更新をロックし，モバイルノードに GKL Ack Message を送る．

4：MLD membership report によってモバイルノードは nMR を通して（S,G）に参加し，MLD membership done によって古い MR（oMR）を通した（S,G）から離脱する．

5：チャンネル移動の後，モバイルノードはグループ鍵サーバに GKL Done Message を送信する．

6：グループ鍵サーバは（S,G）のグループ鍵更新のロックを解き，モバイルノードに GKL Ack Message を送る．

モバイルノードにおける GKL 方式において，ハンドオフによってモバイルノードのアクセスポイントが変わるとき，上記のプロセスを通して，鍵更新が起こる場合には 2 回発生したグループ鍵更新を防ぎ，モバイルノードのハンドオフに伴う処理時間を減らすことができる．

2.8 GKL 方式の問題点

GKL 方式ではハンドオフ時にグループ鍵更新をロックすることで，鍵更新の処理時間を減らすことができる．だが，GKL 方式では 1 つのチャンネルを 1 つのグループとして考えているため，グループのメンバー数を n とすると，処理時間は $O(n)$ となり，メンバー数の増加に比例して鍵更新の処理時間も増えることになる．そのため鍵更新の処理時間が十分減らされているとは言えない．また，1 つのグループ鍵でグループ全体のトラフィックを保護するため，万が一グループ鍵に関する情報が漏れてしまった場合，グループ全体に被害が及ぶ恐れがある．

2.9 本章のまとめ

本章では，既存の鍵管理方式の 1 つである GKL 方式について説明し，その問題点を挙げた．次章では，問題点を改善する手法として，GKL 方式を基にした，階層的な鍵構造を考慮したグループ鍵管理方式を提案する．

第3章

階層的な鍵構造を考慮したグループ鍵管理手法の提案

3.1 本章の概要

本章では，提案するモバイルマルチキャストにおける階層的な鍵構造を考慮したグループ鍵管理方式について説明し，そのメリットとデメリットについて考える．また，提案手法におけるアルゴリズムについて説明する．

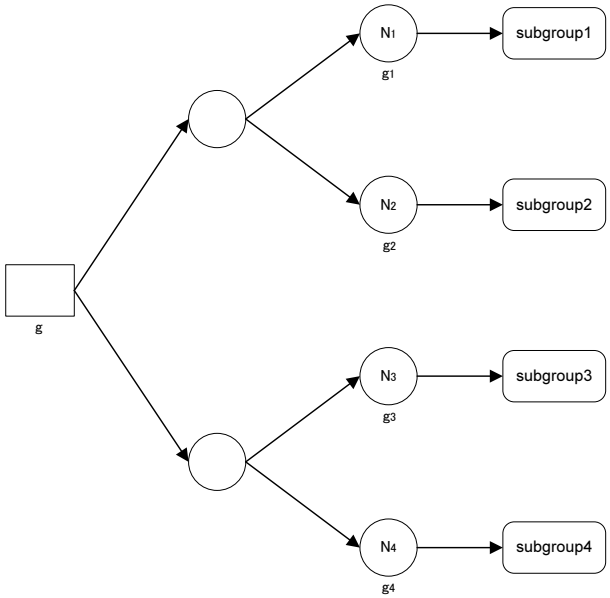

図 3.1: 階層的な鍵構造.

3.2 階層的な鍵構造

階層的な鍵構造は，図 3.1 のようにマルチキャストツリー上に中間ノードを置いて階層化して考える．中間ノードを置くことで，1 つのグループを n 個のサブグループ $N_1, ..., N_n$ に分割し，各 N_i を葉の部分の配置し，N_i には個別のサブグループ鍵 g_i を用いる．このように階層化することで，グループ全体の処理をサブグループで分散できる．また，1 つのサブグループ鍵から他のサブグループ鍵を知ることができないため，安全性の強化も期待できる．

3.3 GKL 方式の修正点

GKL 方式で使われている制御メッセージの修正点について説明する．

3.3.1 チャンネル表記の追加

提案手法ではサブグループの導入に伴い，新たなチャンネル表記である（S,G,Gn）を追加し，これをサブチャンネルと呼ぶことにする．S，G は GKL 方式と同様にソースアドレス，グループアドレスを意味し，Gn は G のグループに含まれるサブグループのアドレスを意味するものとする．

3.3.2 提案手法での制御メッセージ

提案手法では階層化によるサブグループが作られるため，GKL 方式での制御メッセージを使うことができない．そこで GKL 方式での制御メッセージをもとに修正した，提案手法での制御メッセージについて説明する．

GKL Source Message

モバイルソースのハンドオフにおける動作では GKL 方式と同様の動作をするのでこのメッセージによる動作に違いは無いが，メッセージに含まれる情報に変更を加える．もとからある oCoA，nCoA，G,（oCoA,G）の K に加え,（S,G,Gn）の全てのサブグループ鍵 Kn を含む．このメッセージにより，K，Kn の鍵更新をロックする．

GKL Receiver Message 1

このメッセージはモバイル受信者が同一サブグループ内をハンドオフするときに使われる．このメッセージには S，G，oMR，nMR，K に加え，Kn が含まれる．このメッセージにより，K，Kn の鍵更新をロックする．

GKL Receiver Message 2

このメッセージはモバイル受信者が異なるサブグループ間をハンドオフするときに使われる．このメッセージには S，G，oMR，nMR，K に加え,（S,G,Ga）及び（S,G,Gb）の Ka, Kb 以外のサブグループ鍵を含む．このメッセージに含まれている K と Kn の鍵更新をロックする．

GKL Done Message，GKL Ack Message の 2 つのメッセージについては GKL 方式でのメッセージと同様である．

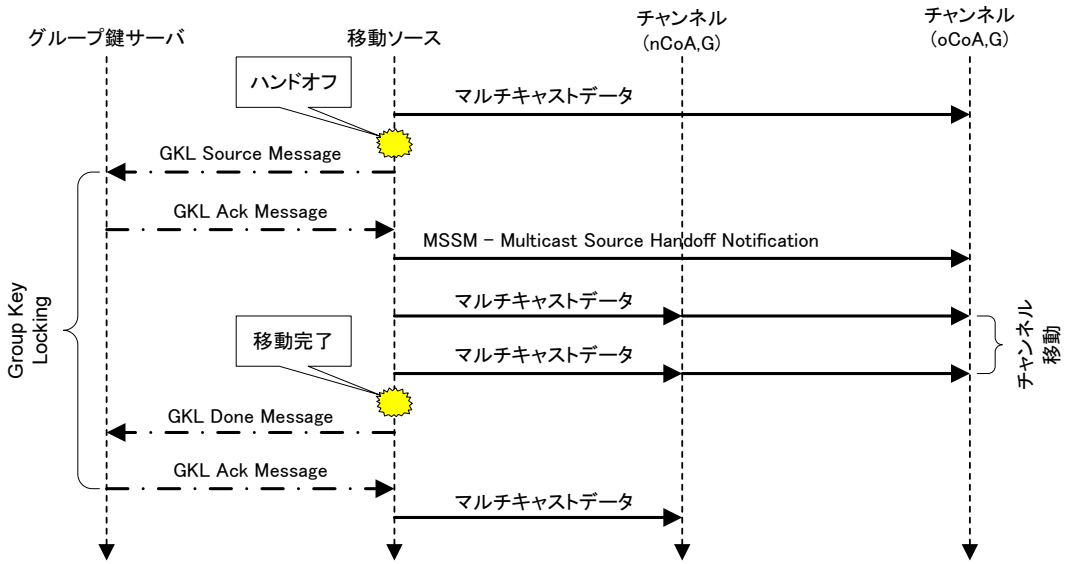

図 3.2: モバイルソースのハンドオフにおける提案手法のプロセス.

3.4 提案手法の動作

提案手法での GKL 方式からの変更点とハンドオフ処理について説明する.

3.4.1 モバイルソースのハンドオフ

モバイルソースにおける提案手法のプロセスを図 3.2 に示す. 処理フローは以下の通りである.

1：ハンドオフが起こるとき，モバイルソースは new Care of Address（nCoA）を設定する.

2：モバイルソースは，GKL Source Message をグループ鍵サーバに送信する.

3：グループ鍵サーバが GKL Source Message に応答すると，（oCoA,G）と（nCoA,G），及び（oCoA,G,Gn）と（nCoA,G,Gn）の鍵更新をロックし，モバイルソースに GKL Ack Message を送る.

4：GKL Ack Message を受けた後，モバイルソースは Multicast Source Handoff Notification をチャンネル（oCoA,G）に送る.

5：Multicast Source Handoff Notification に基づいて，チャンネル（oCoA,G）から（nCoA,G）に移動する.

6：移動が完了すると，モバイルソースは GKL Done Message をグループ鍵サーバに送信する.

7： グループ鍵サーバは,（oCoA,G）と（nCoA,G），及び（oCoA,G,Gn）と（nCoA,G,Gn）のグループ鍵更新のロックを解き，モバイルソースに GKL Ack Message を送る．

モバイルソースのハンドオフの場合，提案手法では GKL 方式と基本的な処理は変わらないが，グループをサブグループに分割することでサブグループごとに分散処理することができるため，ハンドオフでの処理時間を低減できると考えられる．

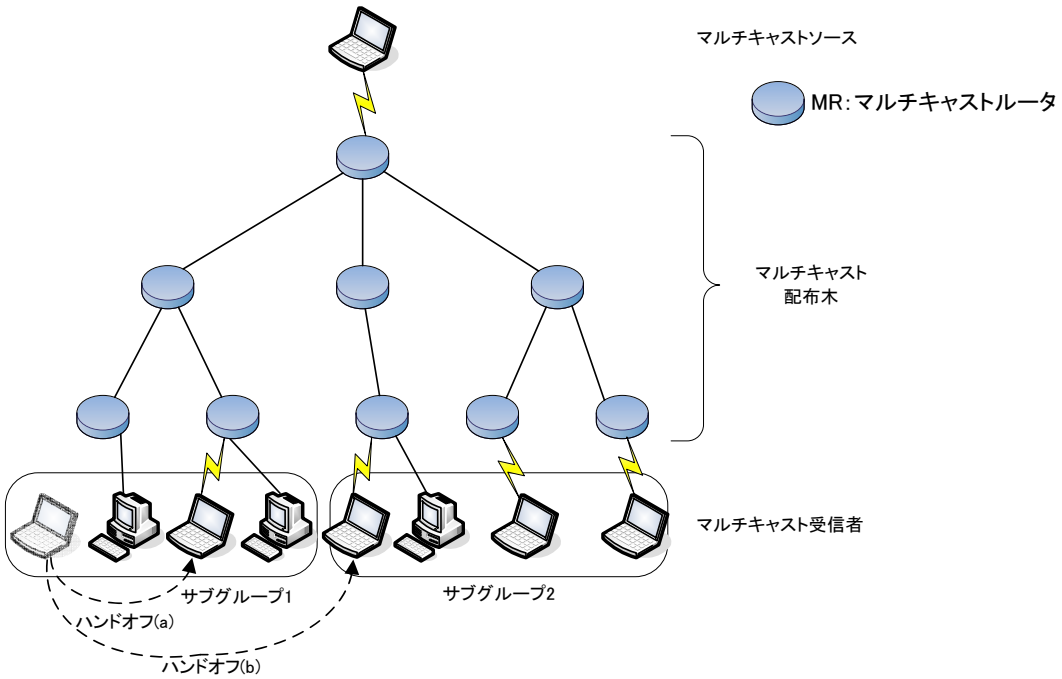

図 3.3: モバイルノードのハンドオフ.

3.4.2 モバイルノードのハンドオフ

モバイルノードのハンドオフの場合，同一のサブグループ内を移動する場合（図 3.3（a））と異なるサブグループ間を移動する場合（図 3.3（b））が考えられる．

同一のサブグループ内のハンドオフ

モバイルノードが同一のサブグループ内を移動する場合の提案手法のプロセスを図 3.4 に示す．処理フローは以下の通りである．

1：モバイルノードのハンドオフが起こるとき，ノードは nMR を設定する．

2：モバイルノードは，GKL Receiver Message1 をグループ鍵サーバに送信する．

3：グループ鍵サーバが GKL Receiver Message1 を受け取ったとき,（S,G）及び（S,G,Gn）の鍵更新をロックし，モバイルノードに GKL Ack Message を送る．

4：MLD membership report によってモバイルノードは nMR を通して（S,G）に参加し，MLD membership done によって oMR を通した（S,G）から離脱する．

5：チャンネル移動の後，モバイルノードはグループ鍵サーバに GKL Done Message を送信する．

6：グループ鍵サーバは（S,G）及び（S,G,Gn）のグループ鍵更新のロックを解き，モバイルノードに GKL Ack Message を送る．

モバイルノードの同一サブグループ内のハンドオフの場合，モバイルソースの場合と同様に，サブグループごとの分散処理によってハンドオフでの処理時間時間を低減できると考えられる．

異なるサブグループ間のハンドオフ

モバイルノードが異なるサブグループ間を移動する場合のプロセスを図 3.5 に示す．処理フローは基本的に同一サブグループ内のハンドオフと同じである．しかし，モバイルノードのハンドオフに伴いサブグループのメンバーに変化があるので，モバイルノードは処理 2 において GKL Receiver Message2 を送信し，グループ鍵サーバは処理 3 において全サブグループの鍵更新をロックするのではなく，GKL Receiver Message2 の情報を元に，変化のないサブグループの鍵更新をロックし，変化のあるサブグループにおいてはロックしない点が同一サブグループ内のハンドオフとは異なる．

異なるサブグループ間のハンドオフの場合，新しいサブグループと古いサブグループの 2 つのサブグループでメンバー変更があるため，2 つのサブグループ鍵の更新が必要となってしまうが，全体的なグループメンバーは変わっていないため，全体のグループ鍵を変更する必要はない．しかし，2 つのサブグループ鍵を更新するので，その分処理時間に遅れが出ることが考えられる．

サブグループをどのように形成するかによって，モバイルノードが異なるサブグループ間をハンドオフする可能性やサブグループごとのメンバー数が変わってくるので，サブグループの形成方法が重要であると考えられる．

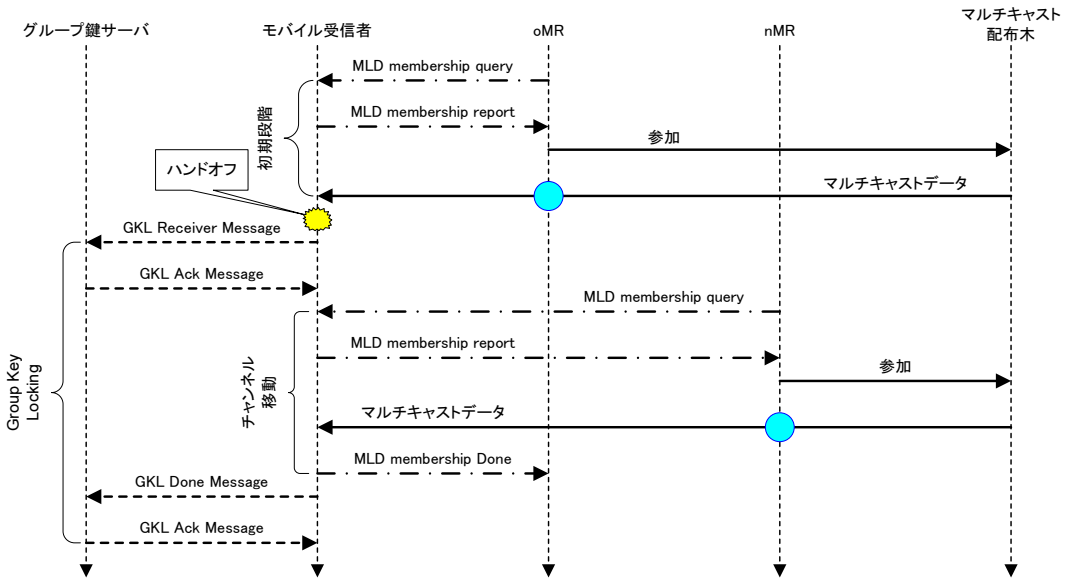

図 3.4: モバイルノードにおける提案手法のプロセス（同一サブグループの場合）.

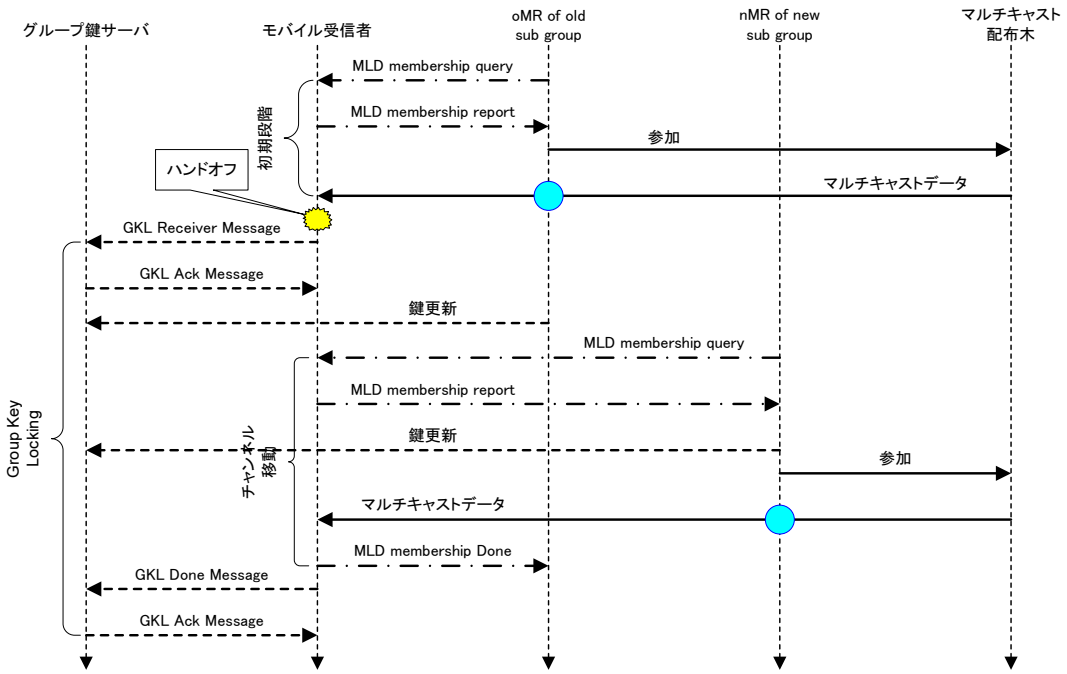

図 3.5: モバイルノードにおける提案手法のプロセス（異なるサブグループの場合）.

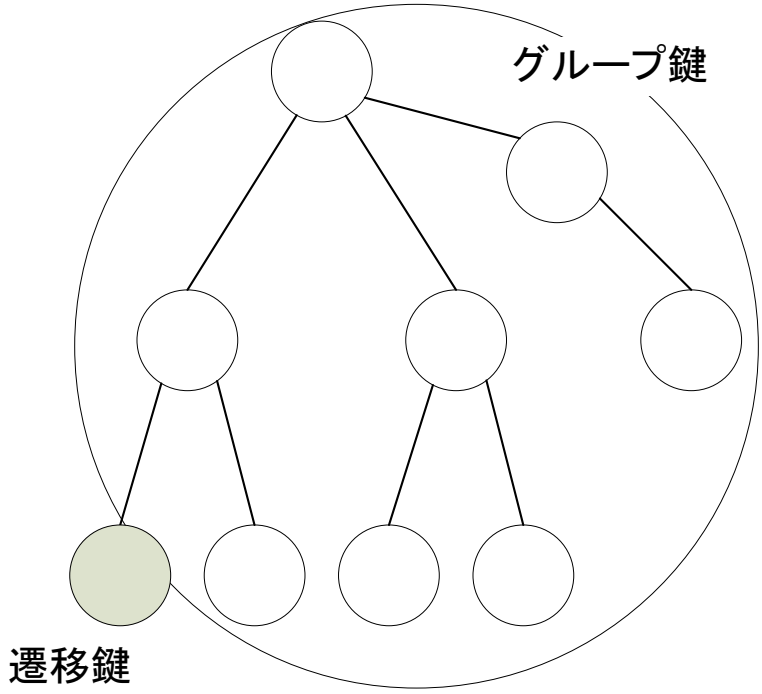

図 3.6: TKS のグループ構造.

3.5 提案手法における鍵更新処理

提案手法において鍵更新が必要となる場合（図 3.3（b））の処理方法について考える.

3.5.1 Transition Key Scheme（TKS）

メンバーの参加によって鍵更新するのではなく，一定の間隔で鍵更新を行う手法，TKS[1] を用いて鍵更新処理を行う．TKS では新しいメンバーがグループに参加する場合，遷移鍵と呼ばれる個別の鍵を用いて，その親と個別のセキュアチャネルを構築する．そして鍵更新が発生すると，新しいメンバーは新たなグループ鍵を受け取って遷移状態を終了し，これ以降，遷移鍵，及び個別のセキュアチャネルを使用しない（図 3.6）.

TKS ではグループ管理者は定期的にグループ鍵の更新を行い，全グループメンバーに送る．またグループ鍵更新時，遷移状態にあるメンバーは新たなグループ鍵を受け取ると，親とのセキュアチャネルを終了する．定期的な鍵更新プロセスでは遷移状態にあるメンバーの分だけ遷移鍵，セキュアチャネルが必要となるが，鍵更新間隔を短くすることで遷移状態メンバーを少なくすることができる．しかし鍵更新間隔を短くすると鍵更新処理が増えてしまう.

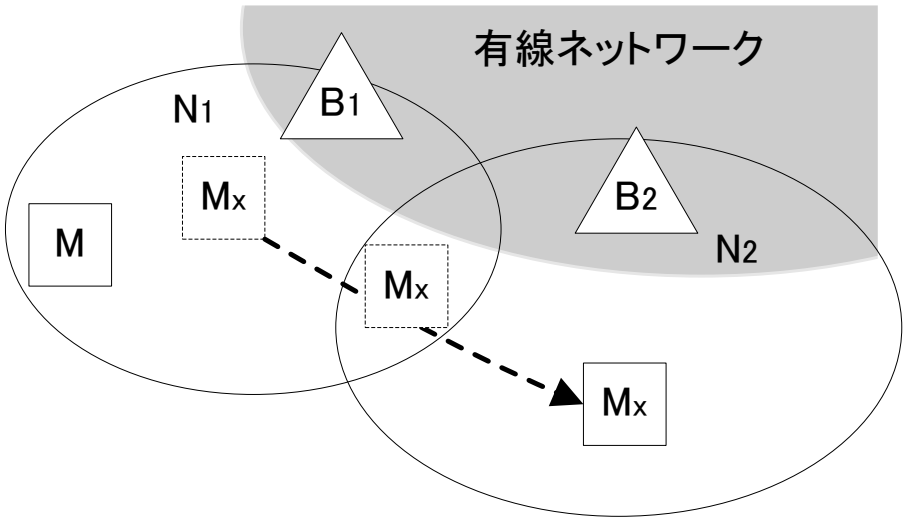

図 3.7: KTMM のハンドオフ処理.

3.5.2 Key Tree in Mobile Multicast（KTMM）

無線領域の重なりを考慮した手法，KTMM[10] を用いて鍵更新処理を行う.

KTMM ではハンドオフするノード M_x は現在のサブグループ N_1 を離脱する前に新たなサブグループ N_2 に参加要求を送る．サブグループ管理者 B_2 はハンドオフするノードを承認し，N_2 のサブグループ鍵を更新して N_2 のメンバーと M_x に送信する．このとき，M_x は 2 つのサブグループ鍵を保有しており，2 つのサブグループに属していることになる．M_x が N_2 へ参加完了後，N_1 から離脱し，B_1 は N_1 のサブグループ鍵を更新することでハンドオフを完了する（図 3.7）.

3.5.3 鍵更新処理手法の比較

TKS を提案手法に導入した場合のメリット・デメリットとして以下の点が挙げられる.

メリット

- TKS による鍵更新処理を用いた場合，通常のグループ鍵更新処理に比べて約 60%の処理時間の低減が可能となる.
- ハンドオフに伴って鍵更新を行わないため，他のノードに対する影響が少ない.

デメリット

- TKS に用いる遷移鍵を新たに導入する必要があるため，扱う鍵の数が増えてしまう.
- TKS はモバイルマルチキャストを想定していないため，提案手法に導入しづらい.

Step1
ハンドオフするノード M_x は新たなサブグループ N_2 に参加要求を送る．

Step2
サブグループ管理者 B_2 はハンドオフするノードを承認し，N_2 のサブグループ鍵を更新して N_2 のメンバーと M_x に送信する．

Step3
M_x は N_2 への参加完了後，N_1 から離脱する．なお N_2 への参加完了から N_1 から離脱までの間，M_x 一時的に 2 つのサブグループに属するので，同一のデータを受け取った場合は一方を破棄する．

Step4
B_1 は N_1 のサブグループ鍵を更新する．

図 3.8: KTMM を用いた提案手法の鍵更新処理.

KTMM を提案手法に導入した場合のメリット・デメリットとして以下の点が挙げられる．

メリット

- ハンドオフするノードが常にグループメンバーに属するため，データを受信し損ねることが無い．
- 新たな鍵などを必要とせず，モバイルマルチキャストにおける鍵管理方式のため提案手法に導入しやすい．

デメリット

- 一時的に 2 つの親が存在し，2 つのデータを受信してしまうため，それを回避する処理が必要となる．
- TKS に比べ処理時間の低減の面では効果が薄い．

TKS を導入した場合，提案手法で必要となるサブグループ鍵に加え，新たに遷移鍵が必要となる点，モバイルマルチキャストを想定していない点から，本手法との相性が悪いため，本研究では導入を見送ることにする．一方，KTMM を導入した場合，モバイルマルチキャストでの鍵管理方式であり，提案手法との相性が良い．また新たな鍵を必要とせず，さらに処理時間の低減の面では効果が薄いものの，無通信時間の低減を図れるといった点から，本手法における鍵更新処理には KTMM を用いることにする．

3.5.4 KTMM の導入

KTMM を提案手法に導入した場合の鍵更新処理を図 3.8 に示す．

提案手法において KTMM を導入した場合と導入しない場合の比較として，サブグループ数を 2，メンバー数を 10〜500 とした時の，図 3.3（b）の処理時間のシミュレーションを行った．シミュレーション結果を図 3.9 に示す．

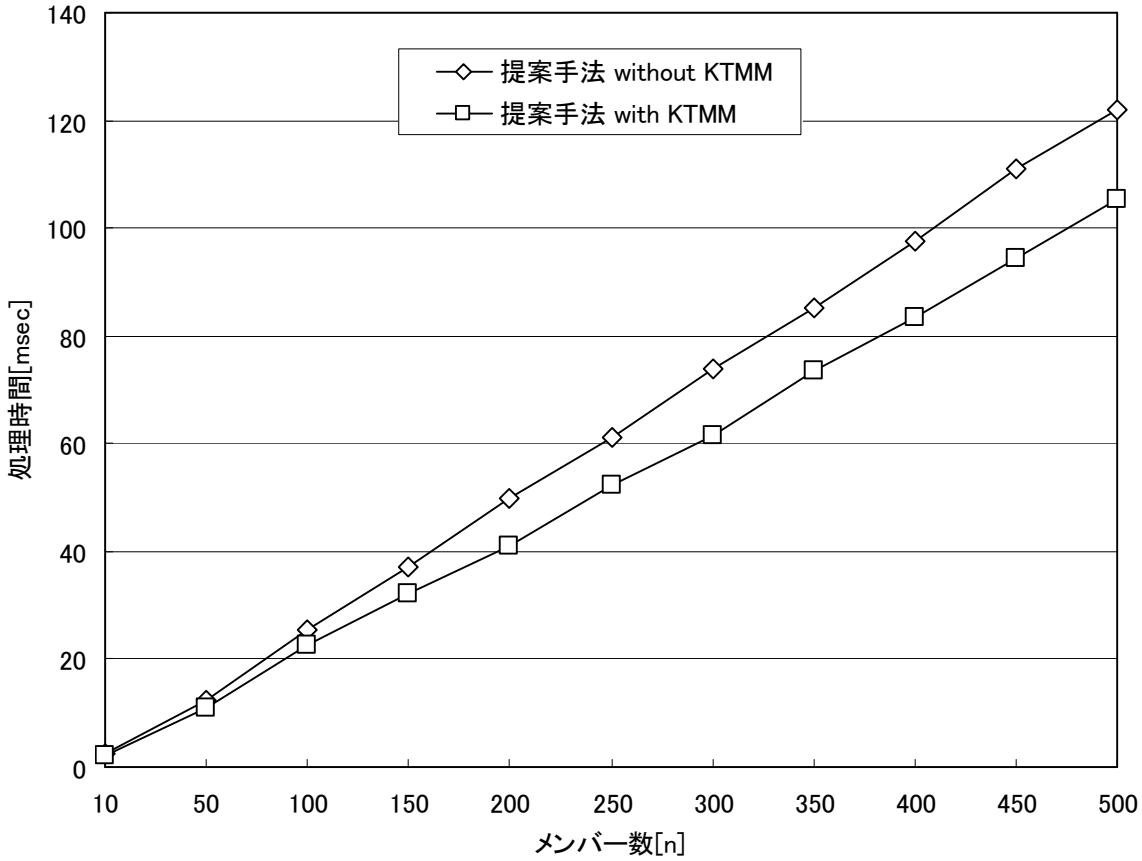

図 3.9: シミュレーション結果.

シミュレーションの結果，KTMM を導入した提案手法は従来の提案手法に比べて平均で 16%優れており，図 3.3（b）における鍵更新処理が僅かながら低減できた．また提案手法の鍵更新処理に KTMM を導入したことで，ハンドオフするノードが一時的にグループから抜けてしまうことが無くなり，無通信時間の低減が可能となった.

3.6 メンバー数に応じたサブグループ数の設定

提案手法において，メンバー数，サブグループ数を変化させた場合の処理時間の変化についてのシミュレーションを行い，提案手法で常に最良の処理時間となるような，メンバー数に応じた最適なサブグループ数を求める．

3.6.1 シミュレーション

提案手法において，メンバー数，サブグループ数を共に変化させ，ノードをハンドオフさせた場合の処理時間のシミュレーションを行った．今回のシミュレーションでは，1 つのサブグループあたりの最大のメンバー数を閾値とし，閾値を超えた場合はそのサブグループを分割して 2 つのサブグループにする方法で行った．閾値を 10，50，100，200 とし，メンバー数を 10〜1000 まで変化させた場合のシミュレーション結果を図 3.10，3.11，3.12 に示す．図 3.10 はメンバー数 10〜1000，閾値=10，50，100，200，図 3.11 はメンバー数 10〜1000，閾値=50，100，200，図 3.12 はメンバー数 10〜500，閾値=10，50，100，200 の場合のシミュレーション結果である．

シミュレーションの結果，メンバー数が少ない場合には閾値が小さい方が，メンバー数が多い場合には閾値が大きい方が処理時間が短いことがわかった．これは，メンバー数が少なく，閾値が大きい場合，鍵構造の階層化がほとんど行われず，サブグループによる分散処理を行えなくなるためである．特にメンバー数が閾値よりも少ない場合は鍵構造の階層化が全く行われず，結果として既存の GKL 方式と同様の処理時間となってしまう．また，メンバー数が多く，閾値が小さい場合，サブグループ数が多くなり，異なるサブグループ間をハンドオフする可能性が高くなることで，サブグループ鍵の鍵更新が発生し，処理時間が長くなると考えられる．

3.6.2 考察

シミュレーションの結果，メンバー数に応じて最適な閾値は異なり，提案手法において常に最良な結果を得るためには，メンバー数に応じて閾値を動的に変更する必要があることがわかった．しかし，実際にはサブグループの分割を行う際の処理時間も考慮する必要がある．特にメンバー数の変動が激しい場合，閾値が小さいとサブグループの分割が多く発生してしまい，処理時間が長くなってしまう恐れがある．

だがシミュレーション結果から，メンバー数が少ない場合において，サブグループ数を変えた場合の処理時間にあまり大きな差は出ていないことがわかっている．そのため，メンバー数 100 まではサブグループ数 2，閾値 50 で十分であり，それ以降は動的に閾値を変更し

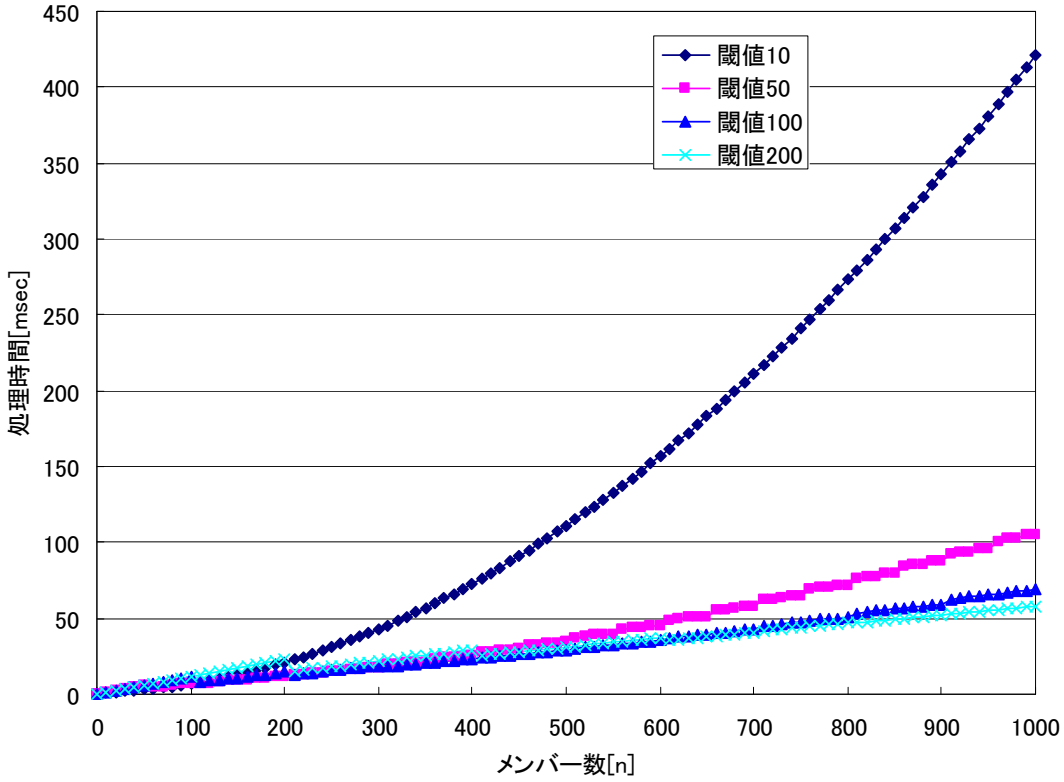

図 3.10: シミュレーション結果（n=10〜1000，閾値=10，50，100，200）.

つつ，サブグループ数を調整することで，提案手法において最良な結果が得られるのではないかと考えられる.

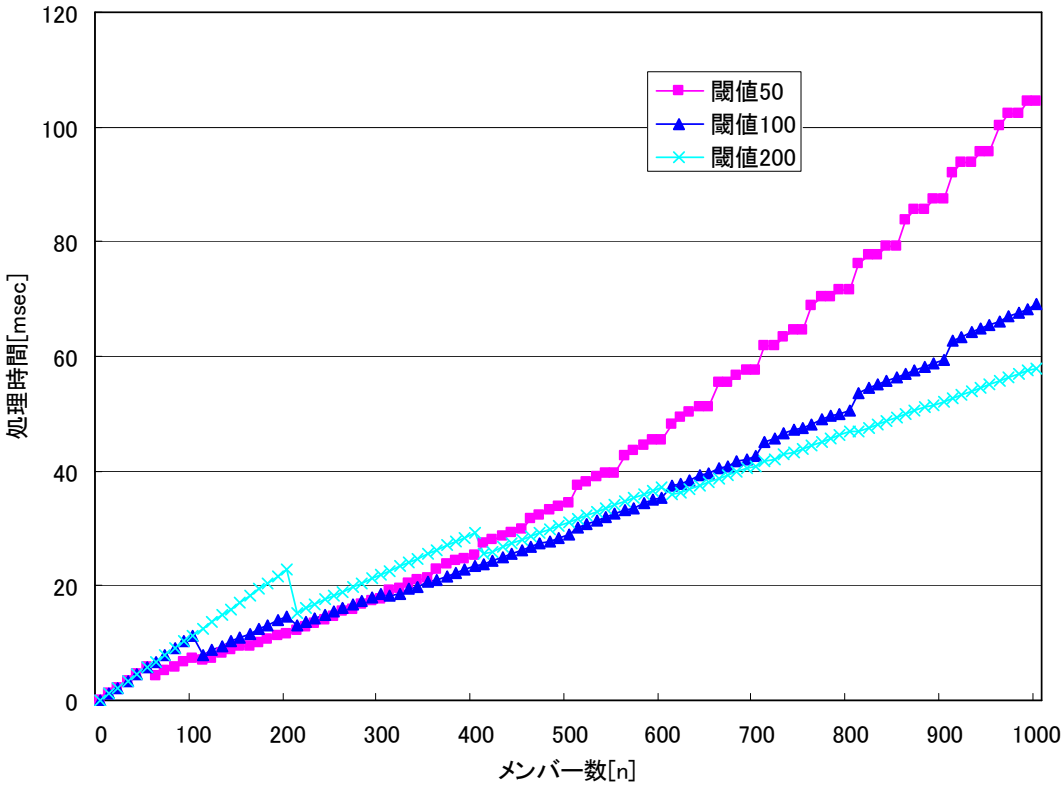

図 3.11: シミュレーション結果（n=10〜1000，閾値=50，100，200）.

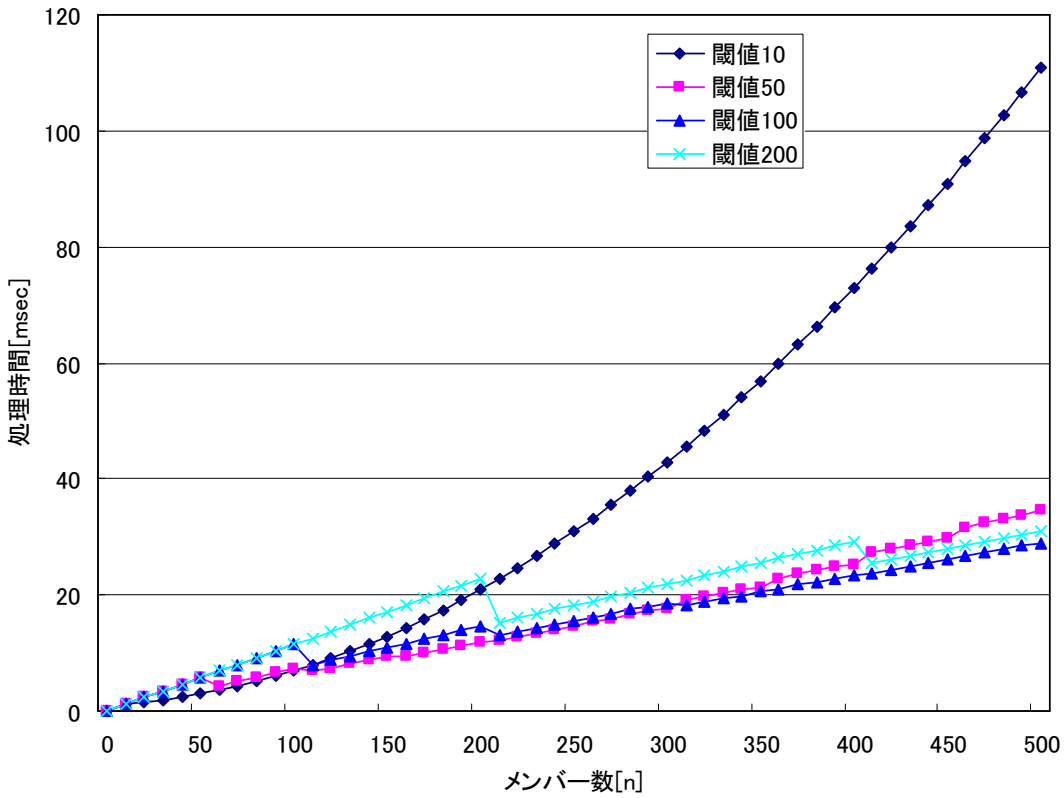

図 3.12: シミュレーション結果（n=10〜500，閾値=10，50，100，200）.

3.7 提案手法の特徴

この提案手法を用いた鍵管理において，考えられるメリットとデメリットについて述べる．

3.7.1 メリット

- グループを階層化し，サブグループを作ることで，参加・離脱処理については変化のあったサブグループに関してのみ考えればよいことになり，鍵更新の処理時間を減らすことができる．
- グループとしての処理を各サブグループで分散処理ができるため，負荷を軽減できる．
- サブグループ数を増やせば 1 つのサブグループ当りのメンバー数は少なくなるため，さらなる処理時間の低減が期待できる．
- KTMM により，鍵更新が発生する場合でも無通信時間の低減ができる．

3.7.2 デメリット

- 扱う鍵が増えることで鍵サーバの負荷が増えることが考えられる．
- 鍵の増加により，暗号化・複合化の段階においてもノードにおける負荷が増えると考えられる．
- サブグループ数を増やすと，鍵サーバの負荷もさらに増える恐れがある．

3.8 本章のまとめ

本章では，GKL 方式の問題点を改善する方式として，階層的な鍵構造を考慮したグループ鍵管理方式を提案した．提案手法では GKL 方式に比べて，処理時間の低減が期待できる．しかし一方で，提案手法では処理時間の低減に伴ってデメリットが発生することも考えられる．

第4章

提案手法の評価

4.1 本章の概要

本章では，第 3 章で提案した手法を実装し，鍵管理について，提案手法と GKL 方式とでシミュレーションによる比較を行い，その評価を行うことで，提案手法の有効性を示す.

4.2 シミュレーション環境

シミュレーションによる比較，及び評価にあたって，ネットワークシミュレーションソフトウェアである Network Simulator 2（ns-2）[13] を使用した．

ns-2 を用いることで様々なネットワークシミュレーションを行うことができるが，ns-2 では鍵管理プロトコルに関しては定義されていない．そのため，グループ鍵管理プロトコルである Group Secure Association Key Management Protocol（GSAKMP）[6] を実装することにする．GSAKMP には実装にあたって要件があるが，本研究では，[6] で「しなければならない（MUST）」,「する必要がある（SHOULD）」と表記された要件のみ実装することにする．また，鍵生成アルゴリズムとして Diffie-Hellman（DH）鍵共有を用いた [2, 9]．

4.3 シミュレーション方法

提案手法におけるメンバー数に応じたサブグループ数については 3.6 節をもとに決定したうえで，メンバー数 50〜1000 のグループにおいて，モバイルノードをハンドオフさせた場合の処理時間について，提案手法と既存研究である GKL 方式とのシミュレーションによる比較を行った．

シミュレーション条件としてはモバイルマルチキャストネットワークを用いた．このネットワーク上でまず，全てのノードが参加要求を送り，マルチキャストグループ，及びサブグループを形成する．また，参加要求に従い，鍵管理サーバはグループ鍵，サブグループ鍵を生成し，それぞれのグループのメンバーに送信する．その後，ノードのうちの 1 つをグループから離脱させ，再度参加させることでハンドオフを引き起こした．このハンドオフによって，ノードが離脱してから，再度参加してマルチキャストパケットを受け取るまでの時間を計測し，その時間をハンドオフにかかる処理時間とした．

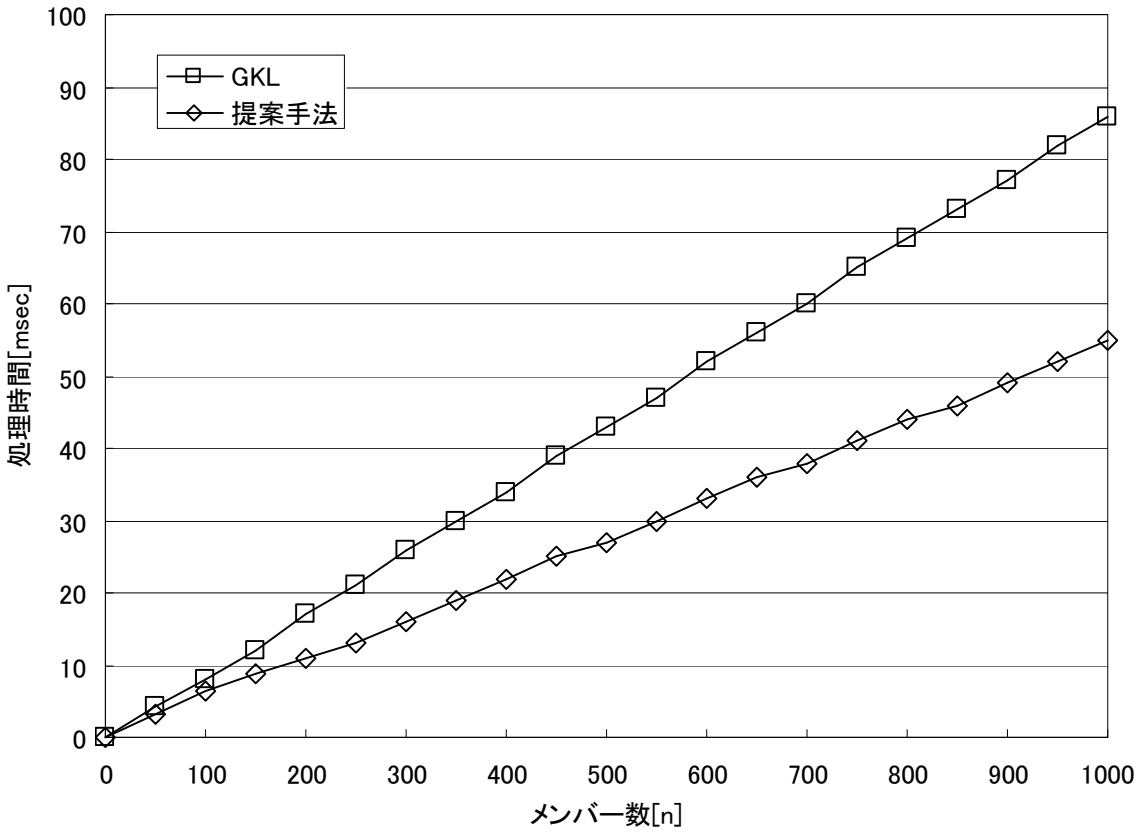

図 4.1: シミュレーション結果.

4.4 シミュレーション結果

シミュレーションの結果を図 4.1 に示す．提案手法は GKL 方式に比べ平均で 34.5%，処理時間の低減が可能となった．メンバー数が多いほど処理時間の差は大きく，メンバー数が 100 の時は 18.7%の低減であるのに対し，メンバー数 850 の時は 37.0%の低減が可能となっている．これは，メンバー数が少ない時にはグループの階層化による効果が小さく，全体の処理時間のうちサブグループの鍵更新処理に関する処理時間が大きいためだと考えられる．

4.5 考察

提案手法は処理時間の面で既存の GKL 方式より大きく優れている．特にメンバー数が多ければ多いほど，グループの階層化による効果が大きく得られることがわかった．またハンドオフするモバイルノードが常にグループに属するため，無通信時間の低減が可能となった．以上のことから，大規模ネットワークでのマルチキャスト通信における鍵管理において，提案手法は非常に有効な鍵管理手法だと考えられる．

一方，本研究では処理時間に着目し，他の指標についてはあまり触れていないが，鍵数の増加によるグループ鍵サーバへの負荷，及び各ノードでの暗号化・復号化処理の複雑化といった課題が残る．まずグループ鍵サーバの負荷に関しては，複数のサーバを用いた鍵の分割管理による負荷分散，鍵生成手法の変更による計算時間の低減などといった対策が考えられる．次に各ノードでの暗号化・復号化処理の複雑化に関しては，同様に鍵生成手法の変更によって，安全性を確保した上での暗号化・復号化処理の単純化といった対策が考えられる．

4.6 本章のまとめ

本章では，提案した手法を実装し，鍵管理について，提案手法と GKL 方式とでシミュレーションによる比較を行った．シミュレーションの結果，処理時間の低減に成功し，提案手法の有効性が示された．一方で提案手法の課題としてグループ鍵サーバの負荷や各ノードでの暗号化・複合化の複雑化といった課題が残る．

第5章

結論

本論文では，既存の鍵管理方式の 1 つである GKL 方式を基にした，モバイルマルチキャストにおける階層的な鍵構造を考慮した鍵管理方式を提案した．マルチキャストにおいては鍵更新における処理時間が重要となるため，提案手法は鍵更新の処理時間の低減を目的としている．

第 2 章「Group Key Locking（GKL）方式による鍵管理とその問題点」では，既存の鍵管理方式の 1 つである GKL 方式について説明し，その問題点を挙げた．モバイルソースもしくはモバイルノードがハンドオフした際，もしグループメンバーの変更が無い場合には，ハンドオフの最中にはグループメンバーに変更があるが，ハンドオフ前後では結果としてグループメンバーは同じになる．通常のマルチキャスト通信では，グループメンバーの変更の度に，グループ鍵の更新をする必要があるが，GKL 方式では，グループ鍵の鍵更新を行わないようにする．この方式によって鍵更新の処理時間を低減できるが，メンバー数を n とすると，処理時間は $O(n)$ となり，メンバーが増えるにつれて処理時間も増えてしまうため，処理時間の低減は十分ではないということが分かった．

第 3 章「階層的な鍵構造を考慮したグループ鍵管理手法の提案」では，第 2 章で説明した GKL 方式を基にして，階層的な鍵構造を考慮したグループ鍵管理手法を提案した．提案手法では 1 つのグループを幾つかのサブグループに分割し，それぞれのサブグループで分散処理することにより，処理時間の低減を図っている．提案手法におけるモバイルノードのハンドオフの場合には，同一サブグループ内のハンドオフか，異なるサブグループ間のハンドオフかによって，サブグループ鍵の更新が必要となる可能性があるが，KTMM の導入によりサブグループ鍵の鍵更新に要する処理時間を平均で 16%低減することができた．また，メンバー数に応じてサブグループ数を変更することで提案手法を用いたハンドオフ処理において最良の結果を得られるようになった．

第 4 章「提案手法の評価」では，実装した提案手法の評価として，ns-2 によるシミュレーションによって，グループメンバー数を変化させた場合について，提案手法と GKL 方式を用いてハンドオフ処理を行った場合の処理時間の比較を行った．シミュレーションの結果，提案手法は GKL 方式に比べ平均で 34.5%処理時間の低減が可能となった．以上の結果から，提案手法の方がハンドオフ処理に要する処理時間が短くなり，提案手法の有効性を示すことができた．

一方，本研究で触れていない指標として，鍵数の増加によるグループ鍵サーバへの負荷，及び各ノードでの暗号化・復号化処理の複雑化といった課題が残る．グループ鍵サーバの負荷に関しては，複数のサーバを用いた鍵の分割管理による負荷分散，鍵生成手法の変更による計算時間の低減，各ノードでの暗号化・復号化処理の複雑化に関しては，同様に鍵生成手法の変更によって，安全性を確保した上での暗号化・復号化処理の単純化といった対策が考えられる．

謝辞

本論文全般にわたり，御指導ならびに御助言を授かった戸川望教授，柳澤政生教授，大附辰夫教授に深く感謝いたします.

また，いつも隣で物欲しげな顔をしている私にお菓子を恵んでくれた本学修士課程の新川将大氏，私のお誘いにホイホイついてきてくれた本学修士課程の橋本識弘氏に深く感謝いたします.

最後に，本論文に関する研究活動全般にわたり支援していただいた戸川研究室，柳澤研究室および大附研究室の皆様に感謝いたします.

参考文献

[1] H. Bettahar, M. Alkubeily, and A. Bouabdallah, "Efficient key management scheme for secure application level multicast," in *Proc. 12th IEEE Symposium on Computers and Communications*, pp. 489–494, Jul. 2007.

[2] W. Diffie and M. E. Hellman, "New directions in cryptography," in *IEEE Transactions on Information Theory*, pp. 644–654, 1976.

[3] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, "Protocol independent multicast - sparse mode (PIM-SM): protocol specification (revised)," *RFC 4601*, Aug. 2006.

[4] B. Han, S. Jung, J. Lee, and T. Chung, "Group key locking scheme in mobile IPv6 networks," in *Proc. The 9th International Conference on Advanced Communication Technology*, pp. 1021–1026, 2007.

[5] 胎中 義也, 山本 幹, "セキュアマルチキャストにおけるネットワーク支援を用いた鍵管理方式," 電子情報通信学会技術研究報告 NS2002-155, pp. 5–8, 2002.

[6] H. Harney, U. Meth, A. Colegrove, and G. Gross, "GSAKMP: group secure association key management protocol," *RFC 4535*, Jun. 2006.

[7] C. Jelger and T. Noel, "Supporting mobile SSM sources for IPv6 (MSSMv6)," in *Proc. IEEE Globecom '02.*, pp. 1693–1697, Nov. 2002.

[8] R. Koodli, "Fast handovers for mobile IPv6," *RFC 4068*, Jul. 2005.

[9] H. Orman, "The oakley key determination protocol," *RFC 2412*, Nov. 1998.

[10] J. H. Roh and K. H. Lee, "Key management scheme for providing the confidentiality in mobile multicast," in *Proc. The 8th International Conference on Advanced Communication Technology*, pp. 1205–1209, Feb. 2006.

[11] S. Setia, S. Koussih, S. Jajodia, and E. Harder, "Kronos: a scalable group re-keying approach for secure multicast," in *Proc. IEEE Symposium on Security and Privacy 2000*, pp. 215–228, 2000.

[12] L. Xu and C. Huang, "Computation efficient multicast key distribution," in *IEEE Transactions on Parallel and Distributed Systems*, 2007.

[13] The Networks Simulator - ns-2, http://www.isi.edu/nsnam/ns/

[14] 総務省情報通信統計データベース，http://www.johotsusintokei.soumu.go.jp/